

Régularisation d'Images Multi-valuées par Convolutions de Lignes Intégrales

David Tschumperlé

GREYC Equipe Image, CNRS UMR 6072
6 Bd du Maréchal Juin, 14050 Caen Cedex. France
David.Tschumperle@greyc.ensicaen.fr

Résumé Nous proposons un nouvel algorithme de régularisation d'images multi-valuées utilisant la technique de convolution de lignes intégrales (aussi appelée LIC [5]). A partir de l'étude d'approches récentes basées sur les EDP de diffusion multi-valuées, nous montrons qu'un processus de régularisation peut être naturellement décomposé en l'élaboration d'une géométrie de régularisation, suivie de l'application d'un lissage local orienté. En exécutant cette dernière étape à l'aide de LIC, nous améliorons de manière significative les résultats obtenus en régularisation d'images, aussi bien en qualité visuelle qu'en temps de calcul. Nous illustrons l'application de notre algorithme à trois problèmes de traitement d'images couleur : le débruitage, l'inpainting et l'interpolation.

Mots clés Images Multi-valuées, Régularisation, Débruitage, Inpainting, Interpolation, EDP's.

1 Introduction

Obtenir une version régularisée d'une donnée dégradée reste un Graal dans le domaine du traitement d'images. L'application directe des algorithmes de régularisation permet par exemple de débruiter des images. Mais ce besoin de régularisation se rencontre aussi fréquemment dans des algorithmes de plus haut niveau mettant en jeu des chaînes d'analyse complexes. Depuis les travaux novateurs de Perona-Malik [9], les EDP de diffusion anisotropes (équations aux dérivées partielles) ont soulevées un fort intérêt pour traiter ce problème : de telles équations sont capables de lisser des images de manière *non linéaire*, ce qui permet de préserver les structures internes significatives présentes dans les images (contours, coins ou autres discontinuités). Ainsi, de nombreux formalismes utilisant les EDP de diffusion ont été proposés dans la littérature pour la régularisation d'images scalaires ou multi-valuées ([1,7,10,11,12,16,19] parmi d'autres). Pratiquement parlant, toutes ces méthodes ont un point commun : elles lisent l'image localement suivant une ou plusieurs directions qui sont différentes en chaque point de l'image. Classiquement, la direction principale du lissage est parallèle aux contours dans l'image.

La définition d'un lissage local correct est donc l'un des points-clés d'un bon algorithme de régularisation. Récemment, les auteurs de [16,19] ont proposés des formalismes très généraux permettant d'élaborer des processus de régularisation adaptés à une géométrie locale de lissage désirée définie à-priori : à partir de l'analyse des structures de l'image (calcul d'un *champ de tenseurs de structure* [19]), un champ de tenseurs de diffusion est calculé. Celui-ci définit le comportement local du lissage désiré. Puis, une ou plusieurs étapes du processus de lissage sont effectuées par l'application d'une EDP de diffusion spécifique.

Dans une première partie, nous résumons tout d’abord ces techniques de régularisation très efficaces. Puis, en gardant à l’esprit l’idée de séparation entre le lissage lui-même et la définition de sa géométrie locale, nous proposons un nouveau formalisme de régularisation d’images multi-valuées basé sur la technique des convolutions de lignes intégrales [5], permettant de lisser une image suivant une géométrie tensorielle définie à-priori. Notre méthode possède deux avantages principaux par rapport aux implémentations par EDP : D’une part, elle préserve mieux les petites structures orientées des images. D’autre part, elle s’exécute jusqu’à trois fois plus rapidement. Pour finir, nous illustrons l’efficacité de notre algorithme LIC, avec quelques résultats d’applications sur des images couleur, pour les problèmes particuliers du débruitage, de l’inpainting et de l’agrandissement par interpolation non-linéaire.

2 Régularisation à base d’EDP

Nous considérons une image multi-valuée bruitée $\mathbf{I} : \Omega \rightarrow \mathbb{R}^n$ ($n = 3$ pour une image couleur) définie sur un domaine $\Omega \subset \mathbb{R}^2$. Nous notons $I_i : \Omega \rightarrow \mathbb{R}$, la composante i de l’image $\mathbf{I} : \forall \mathbf{X} \in \Omega, \mathbf{I}(\mathbf{X}) = (I_1(\mathbf{X}) \ I_2(\mathbf{X}) \ \dots \ I_n(\mathbf{X}))^T$. Les algorithmes de régularisation proposés dans [16,19] consistent à calculer dans une première phase le champ lissé de *tenseurs de structure* $\mathbf{G}_\sigma = \mathbf{G} * G_\sigma$, où G_σ est un noyau gaussien 2D (de variance σ) et $\mathbf{G} : \Omega \rightarrow \mathbb{P}(2)$ est le champ des matrices positives et symétriques suivantes : $\forall \mathbf{X} \in \Omega, \mathbf{G}(\mathbf{X}) = \sum_{i=1}^n \nabla I_i(\mathbf{X}) \nabla I_i^T(\mathbf{X})$.

Les auteurs de [6,19] ont remarqués que $\mathbf{G}_\sigma(\mathbf{X})$ est un bon estimateur de la géométrie multi-valuée locale de \mathbf{I} en \mathbf{X} : ses valeurs propres λ_1, λ_2 donnent les variations vectorielles (couleur) des structures locales de l’image, tandis que l’orientation de ces structures (contours) est donnée par les vecteurs propres $\mathbf{u} \perp \mathbf{v}$ de \mathbf{G}_σ . La convolution de \mathbf{G} par G_σ permet une estimation plus cohérente de la géométrie locale multi-valuée.

A partir de cette simple mesure géométrique \mathbf{G}_σ , les auteurs de [16,19] élabore un champ de tenseurs $\mathbf{T} : \Omega \rightarrow \mathbb{P}(2)$ qui va définir le comportement local du lissage désiré :

$$\forall \mathbf{X} \in \Omega, \quad \mathbf{T}(\mathbf{X}) = f_1(\lambda_1, \lambda_2) \mathbf{u}\mathbf{u}^T + f_2(\lambda_1, \lambda_2) \mathbf{v}\mathbf{v}^T \quad (1)$$

où \mathbf{v} est le vecteur propre principal de $\mathbf{T}(\mathbf{X})$. Intuitivement, f_1 et f_2 définissent la force du lissage qui va être effectué le long des deux directions \mathbf{u} et \mathbf{v} , au point \mathbf{X} . Le lissage lui-même est finalement réalisé en faisant évoluer l’une des deux EDP suivantes :

$$\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{T} \nabla I_i) \quad [19] \quad \text{ou} \quad \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T} \mathbf{H}_i) \quad [16]$$

où \mathbf{H}_i est la matrice hessienne de $I_i(\mathbf{X})$. La comparaison de ces deux équations est discutée dans [16] et sort du cadre de cet article. Néanmoins, l’idée importante qui se cache derrière ces deux méthodes est de baser la régularisation sur l’élaboration d’une *géométrie de lissage local, définie par un champ \mathbf{T} de tenseurs de diffusion*. Notons que \mathbf{T} est généralement mis à jour pendant l’application de l’EDP (après une ou plusieurs itérations de lissage). L’application de telles méthodes est très coûteuse en temps de calcul : \mathbf{I} est régularisé petit à petit car les évolutions d’images par EDP agissent de manière très locales, même avec des schémas numériques récents [18].

3 Régularisation à base de LIC

Pour accélérer le processus de lissage en lui-même, nous proposons de remplacer l’application des EDP de diffusion par une nouvelle méthode plus rapide basée sur des LIC. Nous conservons par ailleurs la première étape des processus par EDP, c’est à dire le calcul (1) du champ de tenseurs \mathbf{T} représentant la géométrie locale du lissage désiré.

3.1 LIC et lissage le long d'une direction

Les LIC (“Line Integral Convolutions” en anglais) ont été historiquement introduits dans [5] comme une technique pour créer un rendu texturé d'un champ de vecteur $\mathbf{w} : \Omega \rightarrow \mathbb{R}^2$. L'idée générale, exprimée dans un formalisme discret, consiste à lisser une image $\mathbf{I}^{\text{bruit}}$ de bruit pur, en moyennant les valeurs des pixels le long des lignes intégrales de \mathbf{w} , et ce, pour tous les pixels $\mathbf{X} \in \Omega$. Cela peut se transcrire à l'aide d'une formulation continue par :

$$\mathbf{I}_{(\mathbf{X})}^{\text{LIC}} = \frac{1}{N} \int_{-\sigma}^{\sigma} f_{\sigma}(a) \mathbf{I}^{\text{bruit}}(\mathcal{C}_{(\mathbf{X},a)}) da \quad \text{avec} \quad \mathcal{C} : \begin{cases} \mathcal{C}_{(\mathbf{X},0)} = \mathbf{X} \\ \frac{\partial \mathcal{C}}{\partial a}(\mathbf{X}, a) = \mathbf{w}(\mathcal{C}_{(\mathbf{X},a)}) \end{cases} \quad (2)$$

où $f_{\sigma} : \mathbb{R} \rightarrow \mathbb{R}$ est une fonction paire (décroissante sur \mathbb{R}^+ , dont la vitesse de décroissance est proportionnelle à σ , typiquement une gaussienne 1D) et $\mathcal{C} : \Omega \times \mathbb{R} \rightarrow \Omega$ définit la *courbe intégrale* de \mathbf{w} passant par \mathbf{X} et paramétrisée par $a \in \mathbb{R}$ (aussi appelée *ligne de champ*). Le facteur de normalisation est $N = \int_{-\sigma}^{\sigma} f_{\sigma}(a) da$. Le paramètre σ contrôle la longueur du lissage (taille du voisinage utilisé pour le filtrage LIC le long de la ligne intégrale). Intuitivement, un lissage de l'image \mathbf{I} suivant le champ de vecteur \mathbf{w} est effectué par (2).

On peut noter que ce problème de visualisation de champs de vecteurs a été abordé par des approches EDP dans [3,15,16], aboutissant à des équations de type $\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{T} \nabla I_i)$ ou $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T} \mathbf{H}_i)$ avec un champ de tenseurs \mathbf{T} défini par : $\forall \mathbf{X} \in \Omega$, $\mathbf{T}(\mathbf{X}) = \mathbf{w}(\mathbf{X}) \mathbf{w}(\mathbf{X})^T$ (définissant donc un lissage anisotrope mono-directionnel). Ces méthodes EDP et LIC pour la visualisation de champs de vecteurs ne sont pas *strictement* équivalentes, mais elles sont basées sur une même idée : lisser une image localement le long de directions définies par un champ de vecteurs \mathbf{w} . Une étude plus approfondie peut être trouvée dans [17].

3.2 Extension au lissage multi-directionnel

La similarité qui existe entre les techniques LIC et EDP de lissage suivant une seule direction suggère naturellement de remplacer la phase de lissage des EDP de régularisation plus générales, par un processus basé sur des LIC. Evidemment, cela implique que nous devons étendre l'utilisation des LIC (qui ne considèrent qu'une direction *unique* de lissage) à des champs de tenseurs de diffusion plus généraux $\mathbf{T} : \Omega \rightarrow \mathbb{P}(2)$. Nous proposons pour cela de *moyenner plusieurs LIC* partant du même point \mathbf{X} , mais suivant des chemins différents.

Soit \mathbf{U}_{θ} le vecteur $\mathbf{U}_{\theta} = (\cos \theta \quad \sin \theta)^T$. Alors, le vecteur $\mathbf{w}_{(\mathbf{X})}^{\theta} = \mathbf{T}(\mathbf{X}) \mathbf{U}_{\theta}$ vérifie :

- Si $\mathbf{T}(\mathbf{X})$ est isotrope, alors $\mathbf{w}_{(\mathbf{X})}^{\theta} = \alpha \mathbf{U}_{\theta}$.
- Si $\mathbf{T}(\mathbf{X})$ est anisotrope et orienté suivant \mathbf{U}_{θ} , alors $\mathbf{w}_{(\mathbf{X})}^{\theta} \simeq \alpha \mathbf{U}_{\theta}$.
- Si $\mathbf{T}(\mathbf{X})$ est anisotrope et orthogonal à \mathbf{U}_{θ} , alors $\mathbf{w}_{(\mathbf{X})}^{\theta} \simeq \mathbf{0}$.

Autrement dit, plus \mathbf{U}_{θ} représente une “partie” de \mathbf{T} , plus $\|\mathbf{w}_{(\mathbf{X})}^{\theta}\|$ sera important.

Pour un θ donné, nous proposons donc de calculer un LIC partant de \mathbf{X} et dirigé suivant le champ de vecteurs \mathbf{w}^{θ} , puis de moyenner tout ces LIC atomiques pour $\theta \in [0, \pi]$. Cet intervalle est suffisant pour balayer l'ensemble du domaine Ω puisqu'un LIC est calculé à la fois dans les sens avant ($a > 0$) et arrière ($a < 0$), le long des lignes intégrales \mathcal{C}^{θ} . Finalement, notre équation de régularisation qui calcule une version lissée anisotrope $\mathbf{I}^{\text{régul}}$ d'une image multi-valuée $\mathbf{I}^{\text{bruité}}$ s'écrit : $\forall \mathbf{X} \in \Omega$,

$$\mathbf{I}_{(\mathbf{X})}^{\text{régul}} = \frac{1}{N} \int_0^{\pi} \int_{-\sigma \|\mathbf{w}_{(\mathbf{X})}^{\theta}\|}^{\sigma \|\mathbf{w}_{(\mathbf{X})}^{\theta}\|} f(a) \mathbf{I}^{\text{bruité}}(\mathcal{C}_{(\mathbf{X},a)}^{\theta}) da d\theta \quad (3)$$

où N est un facteur de normalisation défini par $N = \int \int f(a) da d\theta$, le champ de vecteurs \mathbf{w} est défini par $\mathbf{w}_{(\mathbf{X})}^\theta = \mathbf{T}_{(\mathbf{X})} \mathbf{U}_{(\theta)}$ et la fonction $f(a) = \exp\left(-\frac{a^2}{2\sigma}\right)$. Le paramètre σ est un paramètre défini par l'utilisateur qui règle la puissance globale du filtre de lissage. Comme précédemment, $\mathcal{C}^\theta : \Omega \times \mathbb{R} \rightarrow \Omega$ représente la ligne intégrale du champ de vecteurs \mathbf{w}^θ passant par \mathbf{X} et paramétrisée par a :

$$\begin{cases} \mathcal{C}_{(\mathbf{X},0)}^\theta = \mathbf{X} \\ \frac{\partial \mathcal{C}^\theta}{\partial a}(\mathbf{X}, a) = \mathbf{w}^\theta(\mathcal{C}_{(\mathbf{X},a)}^\theta) = \mathbf{T}(\mathcal{C}_{(\mathbf{X},a)}^\theta) \mathbf{U}_\theta \end{cases}$$

On peut aisément vérifier que :

- Le principe du maximum est respecté par (3), car seul un moyennage des pixels est effectué.
- Pour les points $\mathbf{X} \in \Omega$ où $\mathbf{T}_{(\mathbf{X})}$ est isotrope, le lissage est également isotrope (le moyennage se fait localement dans toutes les directions du plan), puisque $\forall \theta, \|\mathbf{w}_{(\mathbf{X})}^\theta\|$ est constant.
- Pour les points \mathbf{X} où $\mathbf{T}_{(\mathbf{X})}$ est très anisotrope, le moyennage se fait seulement le long du vecteur propre principal de $\mathbf{T}_{(\mathbf{X})}$.

Ces propriétés, également généralement vérifiées par la plupart des EDP de régularisation anisotropes, assurent que le lissage est effectivement dirigé suivant le champ de tenseurs \mathbf{T} .

4 Applications

La Fig.1 illustre quelques applications possible de notre méthode générale de régularisation basée sur des LIC, ici appliquée sur des images couleurs. L'obtention des résultats est très rapide, comparé aux temps nécessaire pour les techniques EDP équivalentes. En effet, notre algorithme ne nécessite que très peu d'itérations pour achever la régularisation, la technique des LIC prenant en compte des valeurs de pixels qui peuvent être très éloignés du point courant \mathbf{X} . Pour nos expériences, nous avons défini \mathbf{T} avec l'éq.(1) et les fonctions f_1 et f_2 par $f_{1/2}(\lambda_1, \lambda_2) = \frac{1}{(\epsilon + \lambda_1 + \lambda_2)^p}$. où $\epsilon = 0.01$, $p = 0.5$ pour f_1 , et $p = 2$ pour f_2 (dans le cas du débruitage) ou $p = 10$ (dans le cas de l'inpainting et de l'agrandissement). La discrétisation spatiale est basée sur de simples différences finies pour le calcul des tenseurs de structure \mathbf{G}_σ (recalculé à chaque itération), et des sommes discrètes sont utilisées pour l'approximation des deux intégrales de (3). La discrétisation de θ est assez large ($d\theta = 30^\circ$) tandis que a est approximé plus précisément ($da = 0.2$). Nous avons appliqué notre équation (3) pour :

- **Le débruitage d'images couleur** (Fig.1a,b,c) : Seulement une itération de notre algorithme est nécessaire. Notez que le bruit est supprimé tandis que les structures orientées de l'image sont bien préservées.
- **L'inpainting d'images couleur** (Fig.1d) : Pour cette application, nous avons appliqué (3) seulement sur les points à l'intérieur d'un masque défini par l'utilisateur (ici ayant une forme de damier). 100 itérations ont été nécessaires pour obtenir le résultat. Une limitation de notre méthode d'inpainting repose sur le fait qu'aucune texture n'est reconstruite.
- **L'interpolation non-linéaire pour l'agrandissement** (Fig.1e) : Ces deux exemples montrent comment notre technique de régularisation par LIC (3) peut être utilisée pour agrandir des images de manière non-linéaire. Des agrandissements linéaires de petites images sont traitées par (3) tout en forçant la valeur des pixels "connus" à rester inchangées.

D'autres résultats, ainsi que l'exécutable de l'algorithme peuvent être trouvés à :

<http://www.greyc.ensicaen.fr/~dtschump/greycstorage/>

5 Conclusion

L'algorithme de régularisation d'images multi-valuées que nous avons proposé est générique et très efficace. Il peut s'adapter à un grand nombre de problèmes en traitement d'images, grâce à la séparation nette entre le processus de régularisation lui-même et la définition de la géométrie de lissage sous-jacente. Nous prenons en fait le meilleur des deux mondes EDP et LIC, ce qui nous permet d'obtenir des résultats de grande qualité visuelle, en un temps de calcul très court. Notre méthode est très simple à mettre en oeuvre, et son champ d'applications est potentiellement très vaste.

Références

1. G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing : Partial Differential Equations and the Calculus of Variations*, vol. 147 of *Applied Mathematical Sciences*. Springer-Verlag, Jan. 2002.
2. D. Barash. *Bilateral filtering and anisotropic diffusion : Towards a unified viewpoint*. Technical report, HP Laboratories Israel, 2000.
3. J. Becker, T. Preusser, and M. Rumpf. *PDE methods in flow simulation post processing*. *Computing and Visualization in Science*, 3(3) :159–167, 2000.
4. M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger. *Robust anisotropic diffusion*. *IEEE Transaction on Image Processing*, 7(3) :421–432, 1998.
5. B. Cabral. *Imaging vector fields using line integral convolution*. *Comp. Graph. Proc.*, p.263–270, 1993.
6. S. Di Zenzo. *A note on the gradient of a multi-image*. *Comp. Vis., Graph. and Image Proc.*, 33, 1986.
7. R. Kimmel, R. Malladi, and N. Sochen. *Images as embedded maps and minimal surfaces : movies, color, texture, and volumetric medical images*. *International Journal of Computer Vision*, 39(2) :111–129, September 2000.
8. T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
9. P. Perona and J. Malik. *Scale-space and edge detection using anisotropic diffusion*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7) :629–639, July 1990.
10. L. Rudin, S. Osher, and E. Fatemi. *Nonlinear total variation based noise removal algorithms*. *Physica D*, 60 :259–268, 1992.
11. G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.
12. G. Sapiro and D.L. Ringach. *Anisotropic diffusion of multi-valued images with applications to color filtering*. *IEEE Transactions on Image Processing*, 5(11) :1582–1585, 1996.
13. N. Sochen, R. Kimmel, and A.M. Bruckstein. *Diffusions and confusions in signal and image processing*. *Journal of Mathematical Imaging and Vision*, 14(3) :195–209, 2001.
14. C. Tomasi and R. Manduchi. *Bilateral filtering for gray and color images*. *Proceedings of the IEEE International Conference on Computer Vision*, pages 839–846, January 1998.
15. D. Tschumperlé, R. Deriche. *Tensor Field Visualization with PDE's and Application to DT-MRI Fiber Visualization*. *Workshop on Variational and Level Set Method (VLSM'2003)*, Nice/France, October 2003.
16. D. Tschumperlé, R. Deriche. *Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications*. *IEEE Journal of Pattern Analysis and Machine Intelligence*, Vol 27, No 4, pp 506-517, April 2005.
17. D. Tschumperlé. *Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's*. *Research Report "Les Cahiers du GREYC"*, No 05/01. Equipe IMAGE/GREYC (CNRS UMR 6072), Février 2005.
18. J. Weickert, K.J. Zuiderveld, B.M ter Haar Romeny, W.J. Niessen. *Parallel Implementations of AOS Schemes : A Fast Way of Nonlinear Diffusion Filtering*. *International Conference on Image Processing* (3 :396-399), Santa Barbara, Oct 1997.
19. J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, Stuttgart, 1998.



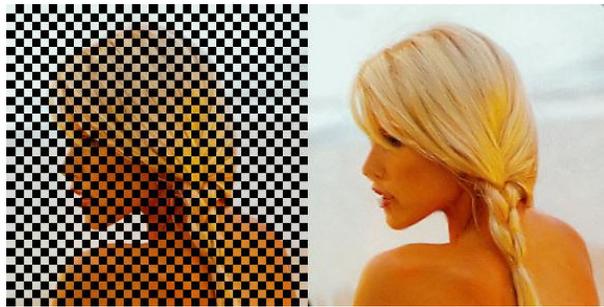
(a) Débruitage d'image couleur avec du bruit granuleux. (1 itération, temps de calcul : 19 seconds)



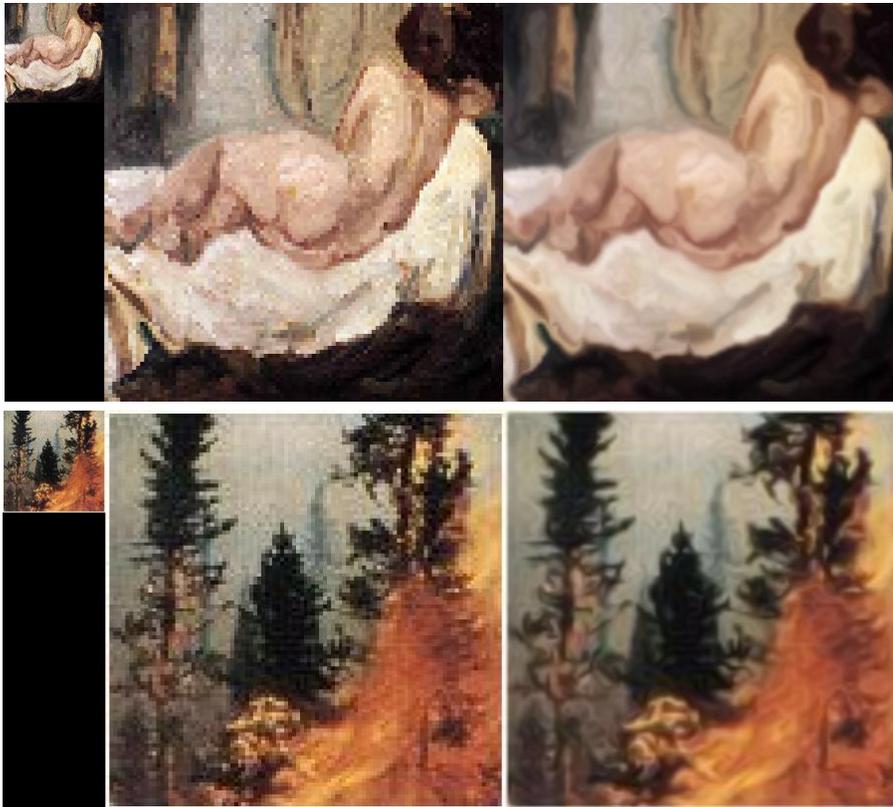
(b) Débruitage d'image couleur avec du moiré. (1 itération, temps de calcul : 11 seconds)



(c) Suppression des artefacts JPG sur une image couleur. (1 itération, temps de calcul : 13 seconds)



(d) Inpainting d'une image couleur. (100 itérations, temps de calcul : 1 minute 26 s)



(e) Agrandissement non-linéaire d'images couleur (1 itération, temps de calcul : respectivement 20 et 5 secondes)
à gauche : Image couleur d'origine, au milieu : Agrandissement par blocs, à droite : Agrandissement par méthode LIC.

Fig. 1. Résultats de notre régularisation basée LIC (3) pour divers problèmes de traitement d'images.